THE LEFT HAND OF SCHOLARSHIP: COMPUTER EXPERIMENTS WITH RECORDED TEXT AS A COMMUNICATION MEDIA

Glenn E. Roudabush,
Charles R. T. Bacon,
R. Bruce Briggs, James A. Fierst, and
Dale W. Isner
The University of Pittsburgh
Pittsburgh, Pennsylvania
and
Hiroshi A. Noguni
The RAND Corporation
Santa Monica, California

To paint a broad though much simplified picture, let us suppose at the outset that scholarship begins with the collection of facts. These facts are of two distinct kinds. The first are observations and they consist, for example, of the results of controlled experiments or observations for field work in the case of science or, perhaps, they are derived from the study of historical documents in the case of history, and so on. The second kind of facts are the *reported* observations, descriptions of phenomena or events, or the theories provided by contemporary scholars. In aggregate, let us refer to the first kind of facts as "data" and the second as "information." From the confluence of these two kinds of facts in the mind of the scholar, new descriptions and theories are born. When he makes these public, then new information is generated.

Scholarship, strictly conceived, is this activity in the mind of the scholar. On its right hand are sources: data and information. On its left are publications: the products of this activity made public. But these two sides of scholarship are closely related. What to one scholar is a publication, to another is information. Every scholar stands both to the right and to the left of every other one.

In our text processing work at the University of Pittsburgh, we look upon our computers and our developing system of programs as a tool designed to extend the abilities of the scholar, on the one hand to collect, sort, and understand information, and on the other to disseminate to others the information that he generates. In other projects and for most of the users at our Center, our computers and systems of programs are seen as a tool to extend the ability to process and analyze data. These systems are, of course, well developed. In analyzing data, one's concern is to reduce, to simplify, and to summarize, preserving only the most significant aspects of the data. While in processing information, we wish to preserve every jot and tittle, allowing no character-

istic of any significance to go unrecorded or untransmitted. Finally, in the research we ourselves do that utilizes natural language text, we come full circle and again use our systems as data processors and analyzers, treating the information we have collected as data.

Figure 1 shows schematically the overall design of our text processing system. Four kinds of input

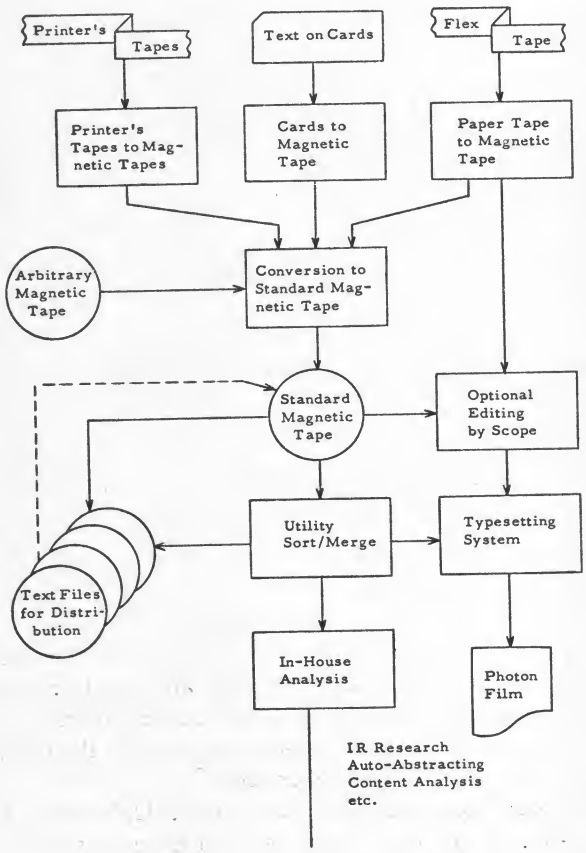

Figure 1. Block diagram of the general text processing system.

are shown. The text on magnetic tape in any arbitrary format may be material obtained from other centers or from any source that produces text on tape. One day this source may include material read by optical character recognition equipment. The printer's control tapes are paper tapes obtained from printers and publishers which were originally used to control some kind of typesetting equipment. We have locally constructed a paper tape reader that will accept 5, 6, 8, 15, and 31-channel paper tape and, through an IBM 1401 computer, write magnetic tape. This work was completed under a Department of Defense Advanced Research Projects Agency grant and has been reported elsewhere.[1,2] The text punched on cards or on Flexowriter-type paper tape would normally represent material prepared at our Center.

The block labeled "conversion to standard magnetic tape" represents the encoding of all forms of natural language text into a particular format according to a schema devised by Martin Kay and Ted Ziehe of the Rand Linguistics Research Group. A relatively complete, but still preliminary description of this format has been published as a Rand Memo.[3] The use of magnetic tape for storage of text and the use of this standard format are prominent in our system and more will be said about this in a moment.

Some source text in exceptionally good condition may, after encoding in this standard form, be ready for distribution to other centers requesting it or for use in our own research. Characteristically, however, some additional processing will be required and this is represented in the block labeled "utility." At the bottom of this figure, our use of text as data is represented. Under "in-house analysis" we have listed information retrieval research, auto-abstracting, and content analysis as examples of this kind of work.

The series of blocks down the right side of Fig. 1 show the normal sequence of operations for photocomposition. Material to be photocomposed will, in most cases, be specifically keyboarded for that purpose. This material will be under good control from the beginning and can go directly into the typesetting system unless it will be used for other purposes as well. Sorting, editing, and other processing will generally not be required so that the conversion to standard format can be bypassed. Both kinds of input to the typesetting system are allowed. An expanded block diagram of the typesetting system itself will be shown in a later figure.

Our system depends to a large extent on the efficient processing of large amounts of natural language text on magnetic tape and this aspect of our system will be described in somewhat greater detail. Magnetic tape is, of course, an economic storage medium and is easily shipped between geographically separated centers. Encoding all text in one standard format becomes important when many different kinds of text from many different sources must be processed and shared. When standardized

input can be expected, a smaller number of general programs can be written and a useful library can begin to be accumulated. The standard adopted must be flexible enough to handle any material one may encounter. The Rand format seems to fill all of our current and anticipated requirements and we have adopted it for our system.

On seven-channel magnetic tape, the minimum unit is a six-bit pattern plus a parity bit. In a one-to-one character representation, only 64 unique characters can be defined. In order to extend the number of different characters that can be represented on tape, either more than one six-bit pattern can be assigned to each character to be represented or else, as in the Rand standard format, some of the available 64 patterns can be used to change the meaning of the patterns that follow them on tape. These mode change patterns or characters are of two kinds: "flags" and "shifts." The flags change the interpretation of succeeding patterns to a new alphabet, while the shifts retain the same alphabet, but mediate changes to, for example, upper case, italics, larger type size, and so on.

Fifteen of the available 64 patterns are permanently assigned as alphabet flags in the Rand system. These 15 patterns along with the blank (octal 60) and a filler character (octal 77) are not a part of any alphabet and their interpretation never changes. There are, then, 47 patterns which can be assigned meanings in each of the 15 alphabets. In each of the 15 alphabets, some of the available 47 patterns will be assigned mode change functions as shift characters. In the Roman alphabet, for example, nine patterns are used in this way. The remaining 38 patterns can accommodate the 26 letters, 10 diacritic marks, and the apostrophe with one pattern left unassigned. Notice that separate alphabets must be used for punctuation, the numerals, and other symbols occurring frequently in the English text.

This encoding system gives a flexible representation of the micro-characteristics of text. Larger units of text, however, have a hierarchical organization which also requires representation. This is accomplished in the Rand system by the "catalog" format. The fundamental unit in this system is the datum, which can be thought of as a manipulable unit of information. A datum may be a text entry consisting of one physical line of text if from a previously printed source, or one sentence, or one word if that is convenient, or it may be a title or a caption from an illustration, or an annotation or description of another datum added at a later time. Each datum belongs to a particular class and at the beginning of each reel of tape following a label record, a map of the corpus is given describing the various classes of material contained in the file. Each datum is coordinated with this map and its proper identification assured by a system of control and label words accompanying every datum. A representation of the Rand encoding system will be shown later in our second typesetting example.

We in Pittsburgh became interested in automatic photocomposition when, in October of 1964, we acquired a Photon S-560 photocomposition machine from the National Institutes of Health. This machine had previously been used by Michael Barnett at the Massachusetts Institute of Technology under an NIH grant. The Photon is an electromechanical device driven by punched paper tape. It consists essentially of a movable glass disk with 1400 characters etched on it and a lens system for projecting these characters onto roll film. The disk can accommodate 16 different type fonts arranged in eight concentric circles or levels around the disk. The paper tape is punched with double character codes, the first giving the character position within disk level and the second giving the escapement for that character. There are additional codes for advancing the film, positioning the film carriage horizontally, affecting lens shifts for size control, and effecting shifts to new disk levels for font changes.

When wereceived the Photon, we also acquired the PC6 system of automatic photocomposition programs developed under the direction of Barnett while he was at M.I.T.[4-8] The PC6 system is typified by the TYPRINT program which requires text containing fixed typesetting control codes as input. These codes are set off from the text by square brackets, which are reserved, and have fixed meanings as shown in the following examples:

[NP] New Paragraph
[DL6] Shift to Disk Level 6 (Highland type face)
[VL2] Leave 2 Blank Lines

In using this system, we soon found that the insertion of fixed codes can be laborious, that changes in format require changes throughout the text, and that many desirable formats are impossible to achieve. We felt that a more flexible and more gen-

erally useful system of programs could be written. We still believe, however, that the PC system was a successful first step toward automatic photocomposition and, in general, the typesetting system we have developed is an outgrowth of our experiences with it.

The input to our system is either magnetic tape in an arbitrary format produced from paper tape punched specifically for typesetting or else magnetic tape in the Rand standard format. The output is again paper tape that will drive the Photon. A schematic diagram of this system is shown in Fig. 2. In this figure, the two forms of magnetic tape input

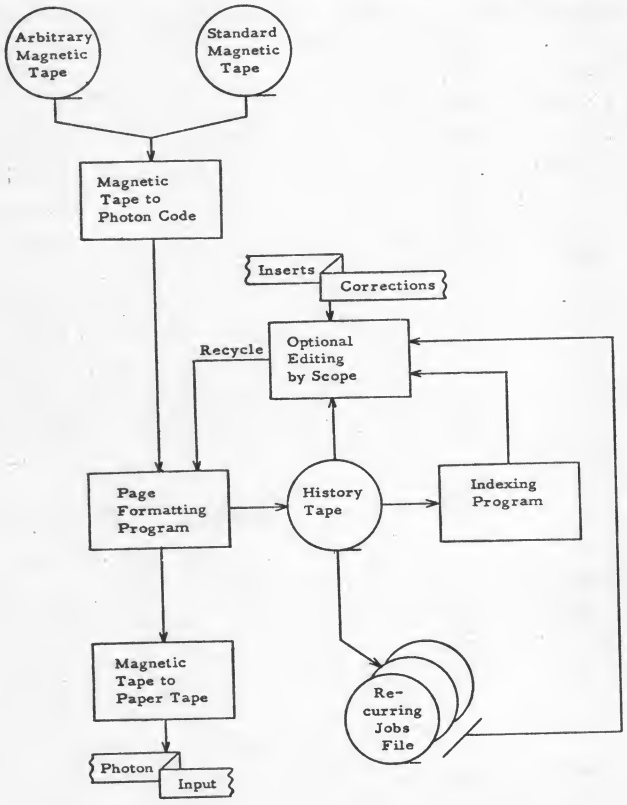

Figure 2. Block diagram of the typesetting system.

are shown at the top. The typesetting program is shown as two separable functions. The first part, which translates text into the double character Photon code, is relatively independent of the second part, but is quite dependent on the particular photocomposition device being used, that is, on the Photon. This part would be largely rewritten if a new piece of equipment were obtained. It is, however, a rather simple and straightforward program. The second part, labeled the "page formatting program," represents a real departure from the PC6 system and other typesetting systems we have seen. In this program, a full page of text is set before outputting is begun.

The page formatting program shows two forms of output. The first is a magnetic tape which contains Photon input that will be converted to paper tape. The other form of output labeled the "history tape," is a magnetic tape containing the original text characters with their associated Photon codes, all of the material added by the page formatting program, page and line numbers, and sufficient parametric information to reset the material exactly as it was originally done. This tape can be recycled through the page formatting program with corrections or additions to the text or simply with changed parameters if the format is to be changed. Since page numbers, tables, captions for figures, titles and subtitles, and so on are all in their proper place on this tape, it can be used as input to a program that produces indices and tables of contents. Finally as shown, this tape might simply be stored for a period of time and then recycled when a new edition is to be set.

This history tape is an important by-product of computerized typesetting and may well be a critical factor in making the adoption of an automatic system economically feasible. This tape is essentially an exact copy of the printed material, less illustrations which cannot be handled in our system, and is a compact, machine-readable counterpart of the standing type that occupies space in some print shops and warehouses. Any material in this file can be simply addressed by page and line number from the corresponding printed document and changes made. If a change is made that affects the remainder of the file, for example an insertion that affects the pagination, all of the file will automatically be corrected.

In designing this system, we came to the conclusion that typesetting control codes in the text to be set are necessary if any format flexibility is to be obtained. They, therefore, appear minimally in our system. We have tried at the same time to ease the burden of keyboarding these codes and of changing their meaning in pre-prepared text by making them entirely arbitrary. The text-dependent codes can be though of simply as markers. The actions to be taken when particular codes are encountered are separately specified as parameters to the system.

These parameters can be inserted anywhere in the text ahead of the markers to which they refer, or they can be punched on parameter cards. If they are keyboarded with the text, they are normally marked off by dollar signs or some other specified reserved symbol. The form of the printd output can be completely changed by changing these parameters with no re-editing of the text itself.

In our system, we wished to include the ability to control as much as possible the layout and final form of the pages in the manuscript. We felt that the deficiencies of other systems in this respect stemmed from their line-by-line typesetting. The attempt to visualize a page by as yet undefined lines is difficult and usually leads to a number of unnecessary trial runs. To ease this difficulty on the programming level, we set full pages. On the conceptual level, we conceive of a page as a collection of subpages or "boxes." A box is a string of fixed text delimited by two markers. The material within a box can be set independently of other material as though it were itself a page and then the box of fixed material placed in its proper position on the page. The box system is recursive so that boxes may be defined within boxes and for most functions, overlapping is allowed.

The parameters used to control the system are of three types: (1) general parameters, (2) text boundary parameters, and (3) box parameters. A list of the general parameters is shown in Fig. 3.

Most of these parameters control the general appearance of the printed output. They include the specification of page size, number of columns on the page, type face, point size, and so on. The parameters specifying running page headers include a provision for incorporating page numbers that are automatically incremented. The last two parameters are provided to make the keyboarding somewhat simpler. The DLIM code allows the specification of any character to mark off parameters when these are included in the text in place of the preset dollar sign. The DEL code allows any character to be specified as a deletion code. It causes a character over which it is typed to disappear from the input string. Only those parameters that are to be different from their preset values need be specified.

The following list of general parameters:

$ PSIZ(8. 5, 11), TFAC(SCOTCH), TSIZ(10), HEAD(Page /1/), COL(3.5, 1.5, 3.5) $

would specify 8½ by 11 inch pages to be set in

SYMBOL	MEANING	NOTES
PSIZ(x,x)	Page SIZe	Page size is width by height.
COL(x,x,x...)	COLumns	Column widths and margins alternate.
JUSV(s)	JUStification-Vertical	Reserved words such as center,spread,etc.
JUSH(s)	JUStification-Horizontal	are used to indicate action desired.
TFAC(n,i,b,B)	Type FACe	n,i,b,B are names of type fonts.
FONT(f)	FONT	Used to indicate italic or bold type.
TSIZ(p)	Type SIZe	Type size is given in points.
BGND(p)	Back GrouND size	Background size is also in points.
TAB(x,x,x...)	TAB	Tab setting measured from left margin.
MWS(p,p)	Minimum Word Spacing	Minimum distance between words.
XWS(p,p)	maXimum Word Spacing	Maximum distance between words.
HEAD(t)	HEADer	Headers may be any string of text,it
LHEAD(t)	Left HEADer	will be set on both pages. LHEAD and
RHEAD(t)	Right HEADer	RHEAD are set on respective pages only.
DLIM(c)	DeLIMiter	Used to surround instructions in text.
DEL(c)	DELetion character	Removes unwanted characters when backspacing.

x is a dimension expressed in inches.

p is a dimension expressed in points.

t is any string of text and may include any boundary or box markers.

c is any keyboardable character and should be one which is not normally used in the text.

n,i,b,B are the names of type faces available. They determine which type face will be used for normal,italic, bold-face and bold-face-italic letters.

s may be one of the following reserved character strings CNT(CeNTer), LFT(LeFT), RGT(RiGhT), SPR(SPRead), BTM(BoTtoM), TOP.

f may be one of the following NOR(NORmal), ITAL(ITALIC), BOLD, or BOLD-ITAL.

Figure 3. List of general parameters.

10 point Scotch with two 3½ inch wide columns separated by 1½ inches. The running heads "Page 1," "Page 2," and so on would print at the top of successive pages. Since the background size and the minimum and maximum word spacing were not specified, reasonable values for these would be computed by the program based on the type size and line length. Hyphenation would occur if the lines could not be justified within these computed limits.

The general form of the text boundary parameters and the box parameters are shown in Figure 4. The text boundary parameters specify a particular arbitrary text marker and a list of actions that are to occur when that marker is encountered in the text. The box parameters specify two particular arbitrary text markers which will delimit fixed strings of text to be treated as a box and a list of actions describing the way material in the box is to be set and the placement of the box on the page. The lists of actions in each of these two parameters can include any of the general parameters or any of the additional actions listed in this figure.

FORM of the TEXT BOUNDARY PARAMETER:

$ AT k)CS(y P,P,P,...,P $

FORM of the BOX PARAMETER:

$ FROM k)CS(y to k)CS(y P,P,P,...,P $

CS is any arbitrary character string.

k is the letter O if the string between the closed and open parentheses specifing the text marker is in octal representation, otherwise it is blank.

y is blank if the marker is not also part of the text, it is S(Save) or SIN(Save IN box) if it is a part of the text.

P may be any general parameter or any of the following.

SYMBOL	MEANING	NOTES
TAB(w)	TAB	Allows indenting to a predefined tab.
SKIP(z)	SKIP	Allows Vertical spacing.
MAR(x,x)	MARgin	Allows margins to be reset.
BSIZ(x,x)	Box SIZe	Specifies dimension of box.
BTAB(w)	Box TAB	Determines horizontal position of box.
BSKIP(q)	Box SKIP	Determines vertical position of box.
UNIT	UNIT	Forces box to be put on one page,i.e., not split.

z may be one of the following reserved symbols NL(New Line), nL(number of Lines), NC(New Column), nC(number of Columns), NP(New Page), nP(number of Pages), nI(number of inches), nPT(number of PoinTs).

q may be the same as z plus TOP BTM(BoTtoM) or CNT(CeNTer) which means the box should be placed at top, bottom or center of the column or page.

w may be a number referring to the nth tab defined plus (for BTAB) LFT(LeFT), RGT(RiGhT) or CNT(CeNTer) which means box should be even with the left, center, or right side of current column or page.

x may be a number (inches) or T (depends on text).

Figure 4. Form of the text boundary and box parameters.

The text markers are defined in these two kinds of parameters as the binary coded decimal equivalent of the character string appearing between the close paren on the left and the open paren on the right. The octal equivalent of the six-bit binary character may also be placed between the parentheses, in which case the letter "O" preceeds the marker specification. A marker may be any string of characters that will not be confounded with text material. They may themselves be a part of the text to be set. If this option is desired, the letter S or SIN (for Save or Save IN) is appended to the marker specification. If S is used, the characters making up the marker are considered to come before the marker or outside the box. If SIN is used, they are considered to come after the marker or inside the box. These conventions give some format control over material that has no keyboarded codes at all.

As a first example of the operation of this system for a straightforward problem, we have taken a part of the "Recent Publications on Computational Linguistics" section of *The Finite String* for June 1965. This monthly newsletter is a publication of the Association for Machine Translation and Computational Linguistics and the short bibliography section has been photocomposed at our Center since April of this year. The procedure we actually use with this material differs somewhat from this description because the text is keyboarded on a Dura Machine 10 at the Rand Corporation rather than at our Center. The differences, however, are minor. In Fig. 5, the Flexowriter hard copy is shown with the parameters appearing at the top of the page. Only

```
$ PSIZ(6.75,10), TPAC(HIGH,HIGH ITAL,CENT BOLD), TSIZ(8),
  XWS(100,100) $


$ FROM  )**(  TO  )**(  TSIZ(9), BSKIP(3L), FONT(BOLD) $
$ FROM  )*ENT(  TO  )*ENT(  SKIP(2L), UNIT $
$ FROM  )[(  TO  )](  FONT(BOLD) $
$ FROM  )/(  TO  )/(  FONT(ITAL) $

**Computational linguistics: Glossaries**

*ENT [Nozaki, A.] "On the Dictionary Preparation,"
manuscript, presented at the U.S.-Japan Seminar on Mechanical
Translation, New York, May 1965. *ENT

*ENT [Lehmann, W.P., and Pendergraft, E.D.]
/Quarterly Progress Report,/
1 November 1964 - 31 January 1965, LRC 65 NSP-23,
Linguistics Research Center, The University of Texas,
Austin, Texas, January 1965. *ENT

*ENT [Nagao, Makoto] "Japanese-English Translation Regarded as
Sentence Generation," manuscript, presented at the
U.S.-Japan Seminar on Mechanical Translation, New York, May 1965. *ENT

*ENT [Otkupshchikova, M.I.] "II simpozium po mashinnomu perevodu"
("2nd Symposium on Machine Translation"), /Nauchno-Tekhnicheskaya
Informatsiya,/ No. 12 (December 1964), pp. 34-36. *ENT

*ENT [Pfafflin, Sheila M.] "Evaluation of Machine Translations
by Reading Comprehension Tests and Subjective Judgments,"
/Mechanical Translation,/ Vol. 8, No.2(February 1965),pp.2-8. *ENT

*ENT [Reitz, Gerhard (ed.)] /Improved Syntactic Flowcharts - Research
Output Format,/ Progress Report No. 9, The Bunker-Ramo Corporation,
Canoga Park, California, 31 March 1965. *ENT

*ENT [Sakai, Toshiyuki] "Procedure for the Analysis of Japanese
Texts," manuscript, presented at the U.S.-Japan Seminar on
Mechanical Translation, New York, May 1965. *ENT

*ENT [Satterthwait, Arnold C.] "Sentence-for-Sentence
Translation: An Example," /Mechanical Translation,/
Vol. 8, No.2 (February 1965), pp. 14-38. *ENT

*ENT [Tosh, L.W.] "Development of Automatic Grammars,"
/Linguistics,/ No. 12 (March 1965), pp. 49-60. *ENT
```

Figure 5. Parameters and text for the *Finite String* example.

general and box parameters are required. The general parameters set the page size to 6¾ by 10 inches, the normal type face to Highland, the italic type face to Highland Italic, the gold type face to Century Bold, the type size to 8 points, and the maximum word spacing to 100 points (to preclude hyphenation). Since the background size and minimum word spacing are not specified, computed values will be used. Four boxes are defined. The first encloses subtitles which are spaced three lines below preceeding material and printed in bold face and somewhat larger type size. The second encloses whole bibliographic entries. The associated actions

cause each entry to be treated as a unit, not to be slit between pages, and a line space is left between them. The third encloses the author's name which is to be set in bold face and the last encloses the title of the publication which is to be set in italics. The photocomposed result is shown in Fig. 6.

This first example was shown to illustrate the simplicity of the system when limited format con-

Computational linguistics: Glossaries

Nozaki, A. "On the Dictionary Preparation," manuscript, presented at the U.S.-Japan Seminar on Mechanical Translation, New York, May 1965.

Lehmann, W. P., and Pendergraft, E. D. *Quarterly Progress Report*, 1 November 1964 - 31 January 1965, LRC 65 NSF-23, Linguistics Research Center, The University of Texas, Austin, Texas, January 1965.

Nagao, Makoto "Japanese-English Translation Regarded as Sentence Generation," manuscript, presented at the U.S.-Japan Seminar on Mechanical Translation, New York, May 1965.

Otkupshchikova, M. I. "II simpozium po mashinnomu perevodu" ("2nd Symposium on Machine Translation"), *Nauchno-Tekhnicheskaya Informatsiya*, No. 12 (December 1964), pp. 34-36.

Pfafflin, Sheila M. "Evaluation of Machine Translations by Reading Comprehension Tests and Subjective Judgments," *Mechanical Translation*, Vol. 8, No. 2 (February 1965), pp. 2-8.

Reitz, Gerhard (ed.) *Improved Syntactic Flowcharts - Research Output Format*, Progress Report No. 9, The Bunker-Ramo Corporation, Canoga Park, California, 31 March 1965.

Sakai, Toshiyuki "Procedure for the Analysis of Japanese Texts," manuscript, presented at the U.S.-Japan Seminar on Mechanical Translation, New York, May 1965.

Satterthwait, Arnold C. "Sentence-for-Sentence Translation: An Example," *Mechanical Translation*, Vol. 8, No. 2 (February 1965), pp. 14-38.

Tosh, L. W. "Development of Automatic Grammars," *Linguistics*, No. 12 (March 1965), pp. 49-60.

Figure 6. The *Finite String* bibliography.

trol is required. Our second example is intended to show a wider range of the possibilities inherent in the system and in particular, the degree of format control that can be obtained. This example consists of the first three pages of an eight-page booklet on postpartum care prepared for the University-affiliated Magee-Womens Hospital in Pittsburgh. After the first two pages, the booklet has a two-column format with captioned illustrations and the author had an exact picture in mind of the way in which each page was to appear. It therefore formed a good test of the formatting capability of our system.

The illustration was prepared in the following steps: (1) the straight text was keyboarded on a Flexowriter without parameters, codes, or markers of any kind; (2) appropriate text boundary and box markers were added using a display scope editing program to be described in a moment; (3) the text with markers was then converted to the Rand standard format; and (4) the typesetting programs were run using this as input. The text editing program used in step (2) is implied by the blocks labeled "optional editing by scope" in both Figs. 1 and 2. The text editor is a general editing program, not specific to the typesetting system, but we have found it very useful in preparing material for photocomposition. We shall give only a brief account of this program here, since a complete description can be found in Bacon.[9]

The text editor program is written for a small Digital Equipment Corporation PDP-4 computer with 4K words of core storage, a cathode ray tube and light pen, a paper tape reader and punch, and a teletype keyboard. This small computer is interfaced into our IBM 7090 giving it access to the tape units, disk file, and core storage of the larger computer. The interface was constructed locally by Russell Ranshaw of our staff. Input to the text editor can be keyboarded directly or read from paper tape or magnetic tape via the interface. Output can be typed, punched on paper tape, or written on magnetic tape.

The text editor continuously displays selected sections of text on the cathode ray tube and editing functions can be performed on the displayed text using the light pen and keyboard. The display is in two parts as shown in Fig. 7. Along the bottom of the screen, stationary symbols are shown which function as push buttons when touched by the light pen. The remainder of the screen is used to display the text being edited. The size and intensity of the characters in the display as well as the vertical and horizontal dimensions of the display itself can be varied. All of the text held in the computer at one

Figure 7. Text editor display.

time can be caused to move down the face of the scope or in the reverse direction with a speed controlled in increments over a wide range of values. This movement of text, its direction, and speed are controlled by the light pen and "push buttons," as are all input and output functions. A complete list of the push-button symbols and their functions is given in the Appendix.

The light pen can be used to place any one of three markers under particular characters in the display. One of these, the cursor, is used to mark a particular point in the text, while the other two, the left and right delimiters, are used to mark off sections of text for deletion or movement. The movement of delimited material to the point marked by the cursor or the insertion of material from the keyboard is controlled by the light pen and push-button symbols. In our illustration, all of the text boundary and box markers were inserted using this program. Figure 8 shows this being in our office by a secretary who has had some experience prepareing material for photocomposition. The text of the illustration as prepared on a Flexowriter is shown at the top of Fig. 9 and then with the required text boundary and box markers inserted at the bottom of this figure.

The next step in the processing of this example was to convert the text with its markers to the Rand standard format. We may suppose that this was done in order to make it available for distribution or to use it for some purpose other than typesetting. A representation of the text in this standard format is shown in Fig. 10. A proper representation would consist simply of a long string of paired octal digits, but that would not illustrate the encoding scheme very well. Here the encoded material is shown on two levels. The upper line shows that shifts and flags while the lower contains the text proper. To the left of each dattum, a text label is shown. This six-character label has the text type indicator as its first character, while the remaining five characters are specific to the entry. This label uniquely identifies the datum. In this figure, the types indicated are T for title, H for heading, A for author, B for body, and C for caption. This information could be used in the typesetting system to control the course of the typesetting process, but in this case, the information is redundant. The flags shown are represented as B for Boundary alphabet, R for Roman alphabet, and P for punctuation alphabet. There are no accepted graphics for the alphabet flags, since they are non-Hollerith (non-printing) six-bit patterns. In the Roman alphabet, shifts are assigned the numerals 1 through 9. The

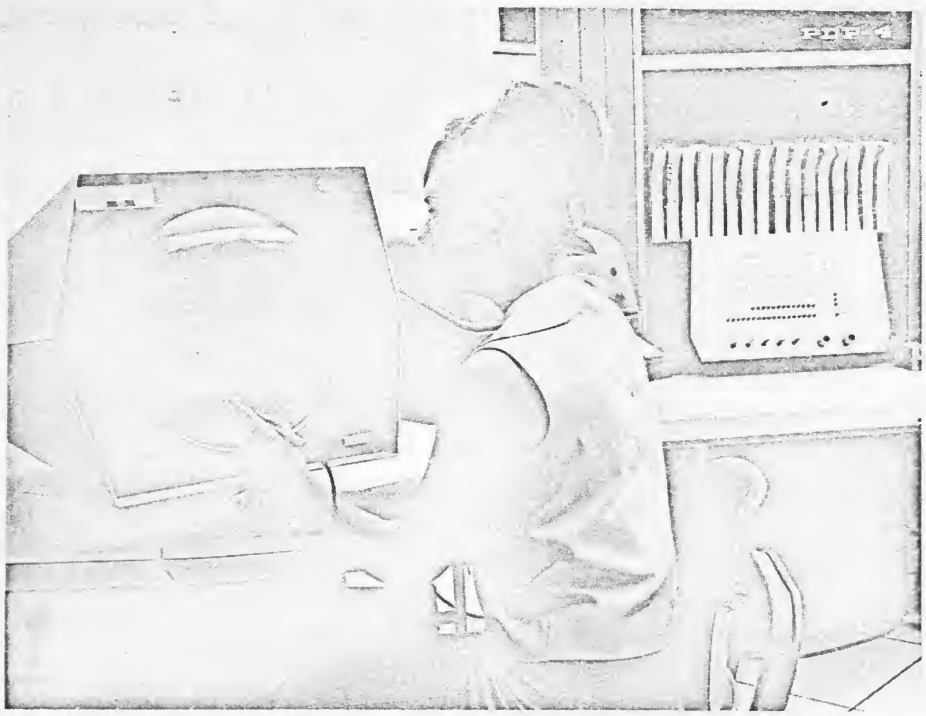

Figure 8. Editing text with scope and light pen.

```
After your baby has arrived...

Magee Womens Hospital
Pittsburgh, Pennsylvania

Prepared by
Barbara Roudabush, R.N.
Illustrated by
Ann Retaichak

Your body returns to normal...

Through a natural process called involution, organs altered by
pregnancy return to normal.

The extra tissues of the uterus and breasts that have built up
during pregnancy are absorbed by the body.

The doctor will measure the progress of this by pressing
lightly on your abdomen and saying how many "finger" widths
the top of the uterus is above or below the navel.

Positions of the uterus after delivery.
```

```
[After][your][baby][has][arrived...]*NP

BOX1 Magee Womens Hospital /
Pittsburgh, Pennsylvania BOX1

*2 Prepared by /
Barbara Roudabush, R.N. *pt
Illustrated by /
Ann Retaichak *2*NP//

B4Your body returns to normal...B4

*5 Through a natural process called involution, organs altered by
pregnancy return to normal. *5

*5 The extra tissues of the uterus and breasts that have built up
during pregnancy are absorbed by the body. *5

*5 The doctor will measure the progress of this by pressing
lightly on your abdomen and saying how many "finger" widths
the top of the uterus is above or below the navel. *5*NC

*B6 Positions of the uterus after delivery. *B6
```

Figure 9. Text before and after editing for the booklet example.

```
T00001:  BR1 9    B R   B R   B R B R      P B
         1 A fter2 lyour2 lbaby2 lhas2 larrived...2N

H00001:  BR1 9    1 9     1 9        BR1 9          P
         3 M agee W omens H ospital 4 P ittsburgh ,

      R1 9            B
        P ennsylvania 3

A00001:  BR1 9          BR1 9      1 9        PR1 9PR1 9P B
         5 P repared by 4 B arbara R oudabush, R . N . 6

      R1 9              BR1 9 1 9          B
        I llustrated by 4 A nn R etaichak 5NA

B00001:  BR1 9                              P B
         7 Your body returns to normal...7

B00002:  BR1 9                                          P R
         8 T hrough a natural process called involution, organs

         altered by pregnancy return to normal.8
                                                      PB

B00003:  BR1 9
         8 T he extra tissues of the uterus and breasts that

         have built up during pregnancy are absorbed by the

         body.8
             PB

B00004:  BR1 9
         8 T he doctor will measure the progress of

         this by pressing lightly on your abdomen and saying how

         many "finger" widths the top of the uterus is above or
              PR    PR

         below the navel.8C
                        PB                                  —

C00001:  BR1 9                                          PB
         9 P ositions of the uterus after delivery.9
```

Figure 10. Representation of the text of the booklet example in standard format.

numeral 1 represents a shift to upper case and the numeral 9 is a shift terminator. The text boundary and box markers appear in the boundary alphabet where the character assignments are arbitrary except for two characters used to delimit sentences and paragraphs. If the parameters had been keyboarded within the text, they would appear either in the Hollerith alphabet or else as text descriptions.

In Fig. 11, the parameters for typesetting this material are shown. The first parameter, $RFORM$, tells the program that the input is in standard format. The general parameters see the page size to 8×5 inches; the type faces used will be Century Italic, Century Bold, and Century Bold Italic; the type size is set to 12 points, and tab positions are set at 1, 2, 3, 4, and 5 inches from the left edge of the page. When the standard format is being used, all text boundary and box markers are assumed to be single characters in the boundary alphabet unless otherwise indicated. In this case, an occurrence in the boundary alphabet of an N causes

```
$ RFORM, PSIZ(8,5), TPAC(CENT,CENT ITAL,CENT BOLD,CENT BOLD ITAL),
  TSIZ(12), TAB(1,2,3,4,5) $

$ AT )N(  SKIP(NP) $
$ AT )4(  SKIP(NL) $
$ AT )C(  SKIP(NC) $
$ AT )6(  SKIP(10PT) $
$ AT )A(  COL(3.75,.5,3.75) $


$ FROM )1( TO )2(  FONT(BOLD ITAL), TSIZ(24), BSKIP(.75),
  BTAB(CNT) $
$ FROM )3( TO )3(  TSIZ(8), BSKIP(CNT), BTAB(CNT) $
$ FROM )5( TO )5(  TSIZ(8), BSKIP(BTM), BTAB(RGT) $
$ FROM )7( TO )7(  FONT(BOLD ITAL) $
$ FROM )8( TO )8(  UNIT $
$ FROM )9( TO )9(  TSIZ(8), BSKIP(BTM), BTAB(CNT) $
```

Figure 11. Parameters for the booklet example.

a skip to a new page, a 4 causes a skip to a new line, a 6 causes a skip down of 10 points, an A causes a two-column format to be initiated (this occurs after the second page), and a C causes a skip to a new column. Each word on the title page is put in a separate box of the same kind. These are set in 24 point bold italics and successive boxes are skipped down ¾ inch and moved to the next tab position. On the second page, the name and address of the hospital are put in a block and centered on the page. The names of the author and illustrator are put in a box and placed in the lower right corner of the page. On page three, the subtitle and each of the three paragraphs are treated as boxes and equally spaced in the left hand column. In the right hand column, space is left for an illustration with the caption centered beneath it. Figure 12 shows these three pages in their final printed form.

With this illustration, the description of our current typesetting system is complete. The typesetting system itself, however, is not now complete, nor will it be until it is abandoned to a dusty completed projects file to rest unused. Some improvements and extensions are planned for the coming months, while others that seem promising will wait for improved hardware. One improvement, for example, will be in the ability of the system to handle tabular material, not only tables of numbers or of words, but also tables of contents and indices. Then again, there is no provision in our system for setting complex mathematical expressions. We have, in fact, no way to represent such forms in a linear string which would allow their efficient reconstruction in two dimensions on paper. This problem, however, is not of great importance to us, since the equipment we have could handle only the simplest formulas.

As our last example has shown, this system has

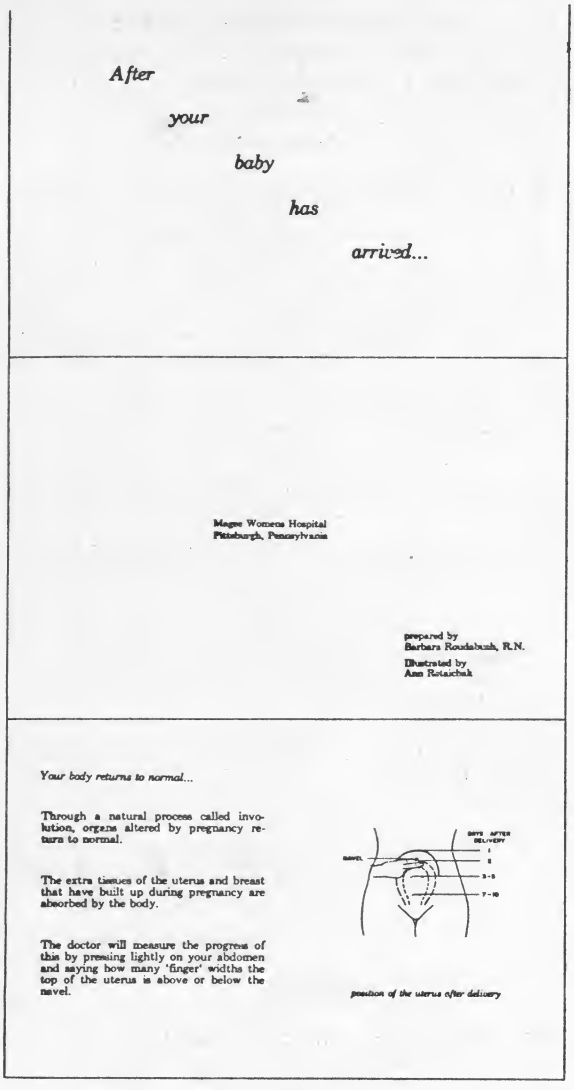

Figure 12. The first three pages of the postpartum care booklet.

some facility in controlling page format in detail. This facility is not as complete nor as easy to use as we would like it to be. We can expect some improvements in the language we use to specify formats and typesetting operations, but significant improvements will wait for new equipment. For typesetting the kind of material exemplified by our booklet illustration, no system will be entirely satisfactory that does not include a manipulable visual display. In such a system, material to be formatted would be punched simply as straight text along with the general parameters and a few text boundary parameters and associated markers, perhaps only marking page boundaries. The system would read the text and display one page at a time on a scope. The author or editor would then move this material about on the face of the scope, changing type size and font at will, until the exact format he wants is obtained. Then, with a push of a button, the page would be written out along with the appropriate codes to set it in that form. The display in such a system would not have to have high graphic arts quality, but the resolution would have to be great enough to provide exact point size and letter spacing representation for the fonts being used.

In the beginning of our discussion, we asserted that the purpose of our text processing system was to extend the abilities of the scholar in performing his work. Whatever else a scholar may do, it seems essential that he be able to: (1) make accurate observations, (2) collect, sort out, and understand the information in his field, (3) integrate his observations with current knowledge to produce new information, and finally (4) make this new information public. We have been concerned with the last of these describing in some detail one particular system intended to aid in the publication of information. The characteristics of this system are derived from our straightforward attempts to use modern computing equipment and programming techniques to duplicate as well as possible the work that is done by printers and publishers. If we are successful, the printed material we produce will be nearly as good as that we are trying to duplicate, but done much faster. If this is the extent of our own scholarly work, then surely we have been unimaginative.

Imagine a system of publication that has the following characteristics. First, a scholar publishes in this system by making his work available on magnetic tape. His publication is then "seen" by other scholars only when a computer has made the decision that his work is both pertinent to and important for some request for information. We assume that the computer's decisions in these matters is less fallable than the scholar's own. Suppose that there are many more subscribers to this system than to any current journal and that the coverage available is just as broad or as narrow as the interests of any individual scholar. Finally suppose that publication in this system is nearly immediate. If this system were in existence, then there would be no further need of scholarly publications in printed form, except perhaps for vanity.

Can the computer do all of this? There are those

of us who think it cannot. But what of the scholar? Can he continue to function for long when the information he must collect and sort out and understand expands exponentially? We may be certain that the scholar will continue to function on some level; that he will continue to generate information. The computer can aid in processing this information. That we already know. The computer alone may not be able to evaluate the importance of a document to some line of investigation, but a computer can hold statistics and the interactions between men and computers may easily generate evaluations. In our research, we are interested not so much in what the computer can do, but rather what the computer and scholar together can do better than either can do alone.

REFERENCES

1. L. Ohringer, *"Computer Input from Printing Control Tapes,"* paper presented at the 16th meeting of the Technical Association of the Graphic Arts, Pittsburgh, June 3, 1964.

2. ———, *"Progress in Computerized Typesetting,"* paper presented at the 17th meeting of the Technical Association of the Graphic Arts, Toronto, Ont., Canada, June 1, 1965.

3. M. Kay and T. Ziehe, *"Natural Language in Computer Form,"* Memorandum RM-4390-PR, RAND Corp. (Feb. 1965).

4. M. P. Barnett, K. L. Kelley and M. J. Bailey, "Computer Generation of Photocomposing Control Tapes, Part 1. Preparation of Flexowriter Source Material," *American Documentation,* vol. 13, pp. 58-65 (1962).

5. E. J. Desautels, *"The Tabprint I System,"* Technical Note No. 33, Cooperative Computing Laboratory, Massachusetts Institute of Technology (May 1963).

6. M. P. Barnett and D. A. Luce, *"The TYPRINT System,"* Technical Note No. 34, Cooperative Computing Laboratory, Massachusetts Institute of Technology (May 1963).

7. ———, ———, and Moss, D. J., *"The RHOPRINT System,"* Technical Note No. 35, Cooperative Computing Laboratory, Massachusetts Institute of Technology.

8. ———, ———, and C. R. Morgan, *"Instructions for Operating the PC6 System,"* Technical Note No. 38, Cooperative Computing Laboratory, Massachusetts Institute of Technology, Jan. 1964.

9. C. R. T. Bacon, *"Text Editing Display,"* Technical Report, The Computation and Data Processing Center, University of Pittsburgh, 1965.

Appendix

THE TEXT EDITOR PUSH-BUTTON SYMBOLS

RUN	Causes the text to be set into motion.
FWD	Causes the motion of the text to be from bottom to top.
REV	Causes the motion of the text to be from top to bottom.
FAS	Accelerates the motion of the text as long as the light pen is held on this symbol.
SLO	Decelerates the motion of the text as long as the light pen is held on this symbol.
HLT	Halts the motion of the text.
MAN	Causes the text to move when the light pen is held on it, if it is otherwise halted, and vice versa.
_C_	Cursor.
_L_	Left delimiter.
_R_	Right delimiter.

The above three symbols control the ability of the light pen to move one or another of the underlines. The symbols themselves vary, with the middle letters C, L, and R remaining constant. An initial letter D shows which of the three underlines may be moved by the light pen, and a final letter D or N tells whether or not the given underline is defined. The cursor is always defined, and hence its symbol always appears as CD or DCD.

TYP	Starts typing on the Teletype the block between the left and right delimiters.
TYH	Causes typing to stop immediately.
DEL	Causes the block between the left and right delimiters to be deleted, operates only if the left and right delimiters are properly defined.
MOV	Causes the delimited block to be moved to the point immediately to the right of the cursor.
CLR	Causes the text area to be cleared.
SPG	This function, the symbol pattern generator, gives rise to a different display. The entire alphabet is displayed and each symbol may be selected for change with the light pen or the Teletype. An enlarged replica of the five-by-seven dot pattern is altered by the light pen to create a new pattern. A light patch in the upper right corner returns the program to the normal text display.

? This is the interlock symbol and it has no function of its own, but serves to activate whatever other function has caused this symbol to be changed. When an interlocked function's symbol is sensed, it is made to appear in place of the question mark. When this new symbol meets the light pen, the function is initiated, and the question mark returns. The CLR function and all input/output functions are interlocked in this way.

IN Reads paper tape and appends the information to the end of the text until the text storage area is almost full, a stop code, or two successive carriage returns have been read.

OUT Causes text to be punched, in Flexowriter format, starting at the beginning of the text and continuing until the character above the cursor has been punched.

DMP Functions the same as the OUT symbol but deletes the text punched.

BIG Causes the letter patterns themselves to be punched and may be used to produce readable titles.

RMT Causes a single 120-character record to be read through the 7090 interface.

WMT Causes a single 120-character record to be written through the 7090 interface.

DMT Causes the entire text area to be written, then cleared, by repetition of the WMT function.

WTM Causes a tape mark to be written, on the output tape.

RWD Rewinds the input and output tape, 7090 logical tapes 2 and 3.

SBC Causes input and output tapes to be logically interchanged. The "SBC" symbol then becomes the "SCB" symbol.

DMR Causes the cursor to be moved to the end of the text, executes the DMP function, and then the IN function.